[AC-5 Red] Teaching Tools for Machine Learning

FINAL REPORT

Course: CS 7993 AI Capstone - W01

Semester: Spring 2025

Website: https://kokou-adje.github.io/Teaching-Tools-For-Machine-Learning/

Github link: https://github.com/Kokou-Adje/Teaching-Tools-For-Machine-Learning.git

Date: April 28, 2025

Team members

Name	**Role**	**Cell Number**	**Email**
Kokou Adje	Project owner	4703583315	adjelaime@gmail.com
Arthur Choi	Project Advisor	7708675309	achoi13@kennesaw.edu

Contents

1.0 Introduction

1.1 Overview

The Teaching Tools for Machine Learning (TTML) is an interactive web-based software implemented to make easier to understand the machine learning algorithms. This platform will focus on the different types of algorithms such as supervised learning, unsupervised learning, and reinforcement learning. TTML represents the bridge between practical applications and theoretical knowledge through an interactive and immersive environment that enables users to explore, discover, investigate, manipulate, and visualize in real-time keys algorithms. This Software Requirements Specification (SRS) document lays out the requirements for this project

1.2 Project Goals

Firstly, the main goal of TTML is to elucidate and demystify the complexity around machine learning concepts through visual and hands on techniques of learning. Secondly, this project seeks to speed up the process of learning to a broader audience interested in artificial intelligence and machine learning particularly. Finally, TTML tries to provide a collaborative learning environment that allows users to share and compare results from other platforms and experiences.

1.3 Assumptions

The design and the implementation of this project are based on certain assumptions. It is assumed that the users of the application have a good connection and browsers able to display and render complex visualization and support a responsive user experience. This assumption is important for the performance and accessibility of the platform. The second assumption is related to users' basic knowledge to comprehend some mathematics concepts such as statistics and algebra. The third assumption guarantees that the users have a stable internet connection able to support data processing and visualization in real-time manner.

2.0 Design Constraints

The implementation of TTML presents some design constraints such as environment, user characteristic, and system requirements described in this section.

2.1 Environment

TTML is implemented as a web-based software that uses modern web technologies which allow users to access the platform through major browsers such as Edge, Firefox, Safari, Chrome without any installation.

TTML's backend uses Python as the main programming language that represents the best and most used language in artificial intelligence ecosystem. Python overflows machine learning libraries including Pandas, Scikit-learn, and NumPy utilized in data manipulation, preprocessing, modeling, and analysis. For the backend, RESTful API will be integrated to establish the communication between the backend and the frontend. This architecture will ease the incorporation of new features. TTML's frontend will be constructed around JavaScript frameworks such as Vue.js or React coupled with visualization libraries such as Plotly.js or D3.js which allow to create dynamic and interactive user interfaces to render in real time complex data visualization. The application will be responsive and adaptable to any screen size and device types like smartphones, tablets, and computers.

The following table summarizes some requirements for the hardware environment.

CPU	RAM(GB)	OS(GB)	External Disk	OS type	Browser
15	512	1 000	500	Windows	Edge Google Chrome
20	1000	500	1000	Linux	Microsoft Edge

2.2 User Characteristics

TTML is implemented for a wide range of users having various levels of skills in machine learning. The main audience of TTML is composed of undergraduate and graduate students in data science, computer science, and professionals seeking to enhance their expertise in machine learning. This platform offers an adaptable and flexible learning environment supporting those different users.

The platform provides an intuitive user interface and easy navigation, and no prior programming skill is required. The platform also gives the opportunity to most advanced users to explore deeply some algorithms.

2.3 System

The design of TTML is based on performance, reliability, and performance. Its architecture utilizes a microservices model allowing individual components to increase independently according to demand. This architecture guaranties that the system can handle many users while providing fast and responsive quality. Google Cloud Platform, exploiting their abilities to ensure high availability and optimal resource utilization, will be used in the backend.

Data processing and algorithms execution will be run on the server side to reduce the requirements related to the client side. The system will integrated advanced caching strategies and data streaming techniques to manage large datasets efficiently and erase latency during data visualization updates.

3.0 Functional Requirements

3.1 Dataset uploading

- The system allows users to upload a dataset.
- The system checks if the format is valid with required columns.

3.2 Raw Data Preview

- The system displays some rows of the uploaded dataset.
- The system gives the possibility to preview or not the dataset.

3.3 Prediction making

- The system predicts the label according to the attributes provided by the users.
- The system allows users to input different kinds of values to have an output.

3.4 Real-time visualization

- The system displays or visualizes the uploaded dataset and the results in real-time.
- Dynamically, the system updates the visualization without needing to reload the page.

3.5 Choice of Algorithm and visualization

- Algorithm selection
    - ✓ Users will choose the desired algorithm between linear regression, k-means clustering, logistic regression.
    - ✓ The description of the algorithm will be displayed.
- Algorithm execution
    - ✓ Users can execute the selected algorithm and dataset.
    - ✓ The result will be displayed in real-time.
    - ✓ Each step of the algorithm execution will be displayed with an explanation.
- Visualization and Interaction
    - ✓ Interactive Plots will be available for each selected and executed algorithm.
    - ✓ Users can zoom the result.

4.0 Non-Functional Requirements

The non-functional requirements are related to security, capacity, usability, and other.

4.1 Security

The primary priority of TTML is security because of the sensitivity of the data manipulated on the platform. The system will utilize a multi-layer technique to secure user information, protect the integrity of data, and guarantee the safety of the platform. User authentication is one of the key security measures of the system. Also, the system will provide Multi-factor Authentication to protect users accounts. To respect data privacy, AES-256 encryption will be implemented to secure dataset uploaded.

4.2 Capacity

The capacity requirement for TTML consists of handling the quantity of information, data, and resources allocated to the system to meet the desired performance. The data's volume, the storage capacity, the hardware, and the database performance need to be examined.

4.3 Usability

The usability requirement involves how easy users can navigate and interact with the system. The user interface will be intuitive, user-friendly, and must be adaptative to any other devices such as computers, Mac OS, android. The application will facilitate the navigation between pages easily and clearly.

4.4 Other

The application will be reliable, extensible, and allow the integration of new features and functionality without reducing enough capacity and performance. The system will be designed with fault tolerance by handling future and potential errors during the software utilization.

5.0 Design Considerations

5.1 Assumptions and Dependencies

The design and the implementation of this project are based on certain assumptions. It is assumed that the users of the application have a good connection and browsers able to display and render complex visualization and support a responsive user experience. This assumption is important for the performance and accessibility of the platform. The second assumption is related to users' basic knowledge to comprehend some mathematics concepts such as statistics and algebra. The third assumption guarantees that the users have a stable internet connection able to support data processing and visualization in real-time manner.

5.2. General Constraints

The implementation of this platform faces various challenges and constraints. One of the difficulties is the browser compatibility where the system must be compatible with major browsers by impacting the frontend implementation. Another constraint is to handle the largest datasets which lower the data processing time. The third constraint is compliance with the regulation which necessitates that the system protects the users.

5.3 Development Methods

The methodology utilized in TTML is a hybrid including Agile methodology and the requirement of the software. This technique is chosen to fix the difficulties and iterative nature of the implementation based on machine learning. The main methodology adopted in this project is the adjustment of the Agile framework by integrating Kanban and Scrum elements. The flexibility required for this project and its iterative development explained this choice.

5.4 System Architecture and Design

The TTML is implemented with a microservices architecture that guarantees flexibility, scalability, and maintainability. The system is divided into the architecture below.

- ✓ Frontend service: is an entry point to the system. The technology used is Streamlit-based to upload CSV files, select parameters, predict and visualize.
- ✓ Backend service: Python for data validation, model training, and output computation.
- ✓ Algorithm execution service: execution of diverse machine learning algorithms based on user selection. Technology utilized are: Python, Panda, NumPy, scikit-learn.
- ✓ Visualization Service: model visualization using Streamlit.

5.5 Data Flow Diagram

Data flow diagram is used to show how data is moving through the system.

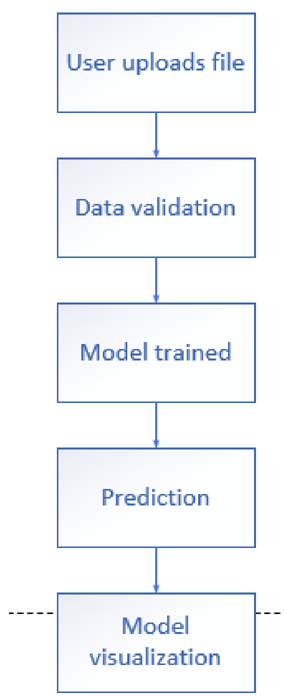


Figure 1 - Data Flow Diagram

6.0 Version Control Plan

For our project TTML, we develop a version control plan using Git and GitHub as remote repository to ensure the project tracking and code management. The detailed plan:

a. Structure of the repository:
    - Main repository: Teaching-Tools-For-Machine-Learning

b. Branching approach:
    - Main: stable code for the production
    - Develop: code for feature implementation
    - Feature: individual feature branch

c. Commit guidelines:
    - Provide a clear and well-descriptive message during the commit
    - Follow the commit convention

d. Process of Pull request:
    - Make a pull request to merge feature branches into develop branch
    - Before merging, make a brief code review

a. Versioning:
    - For release naming, comply with Semantic Versioning
    - In the repository, tag all releases

b. Documentation:
    - Keep updating README.md for each repository

c. Backup
    - Proceed to a backup of the GitHub repositories to a different storage.

7.0 Deliverables

TTML implementation follows the different aspects of software development life cycle (SDLC) and its breakdown is below.

d. Project planning:
    - Project plan
    - Gantt chart

e. Requirements and Analysis:
    - Software Requirements Specification (SRS)
    - Data Flow Diagram

f. Design:
    - Software Design Document (SDD)
    - Database Schema Document (DSD)

g. Implementation:

- Source code (hosted on Github)
- Version control (using GIT)

h. Testing:
   - Test Plan

i. Final Deliverables:
   - Final Report Package
   - Source Code Repository(https://github.com/Kokou-Adje/Teaching-Tools-For-Machine-Learning.git)
   - Website (accessible at https://kokou-adje.github.io/Teaching-Tools-For-Machine-Learning/)
   - Video Demo
   - C-Day Application/Submission

Milestone Events (Prototypes, Draft Reports, Code Reviews, etc)

The following are the milestone events for TTML project and their approximative dates:

#1 Project Planning – By January 26, 2025
- Project Plan
- Gantt Chart

#2 Requirements Analysis and Design (RAD) – By February 02, 2025
- Software Design Document (SDD)
- Database Schema Document (DSD)

#3 Prototype Presentation – By March 16, 2025
- Presentation Slides
- Peer Review

#4 Final Report Package Preparation – By March 24, 2025
- Final Report Package Draft
- Website (building pending)

#5 C-Day Submission – By April 7, 2025
- Test Plan
- Development documents

#6 Final Report – By April 28, 2025
- Final Report Package
- Source Code Repository(https://github.com/Kokou-Adje/Teaching-Tools-For-Machine-Learning.git)
- Website (accessible at https://kokou-adje.github.io/Teaching-Tools-For-Machine-Learning/)
- Video Demo

Meeting Schedule Date/Time

The meeting is scheduled for Saturday and Sunday every week.

Collaboration and Communication Plan

Communication	—	Cellphones (Call/Text) / Microsoft Teams / GroupMe
Collaboration	—	Discord (Mandatory unless another tool is authorized by Perry)
Version Control	—	GitHub

8.0 Summary

The Teaching Tool for Machine Learning project delivers successfully a web platform implemented to enhance teaching and learning AI concepts via hands-on experimentation. TTML tries to close the gap between practical application and theoretical knowledge by allowing users to upload their own dataset in respect to the required format, select the attributes, predict the label or the output, and then visualize the model in real-time. The implementation of this project follows the SDCL and agile methodology including requirements gathering, design, development, and testing. TTML has progressed by making machine education more engaging, accessible, easy to understand, and interactive. TTML, with its current progress and the incoming plans, has a significant chance to become an important tool for learners such as students, instructors, and professionals.

9.0 Appendix

9.1 Gant Chartt

Project Name: [AC-5 Red] Teaching Tools for Machine Learning
Report Date:April 28, 2025

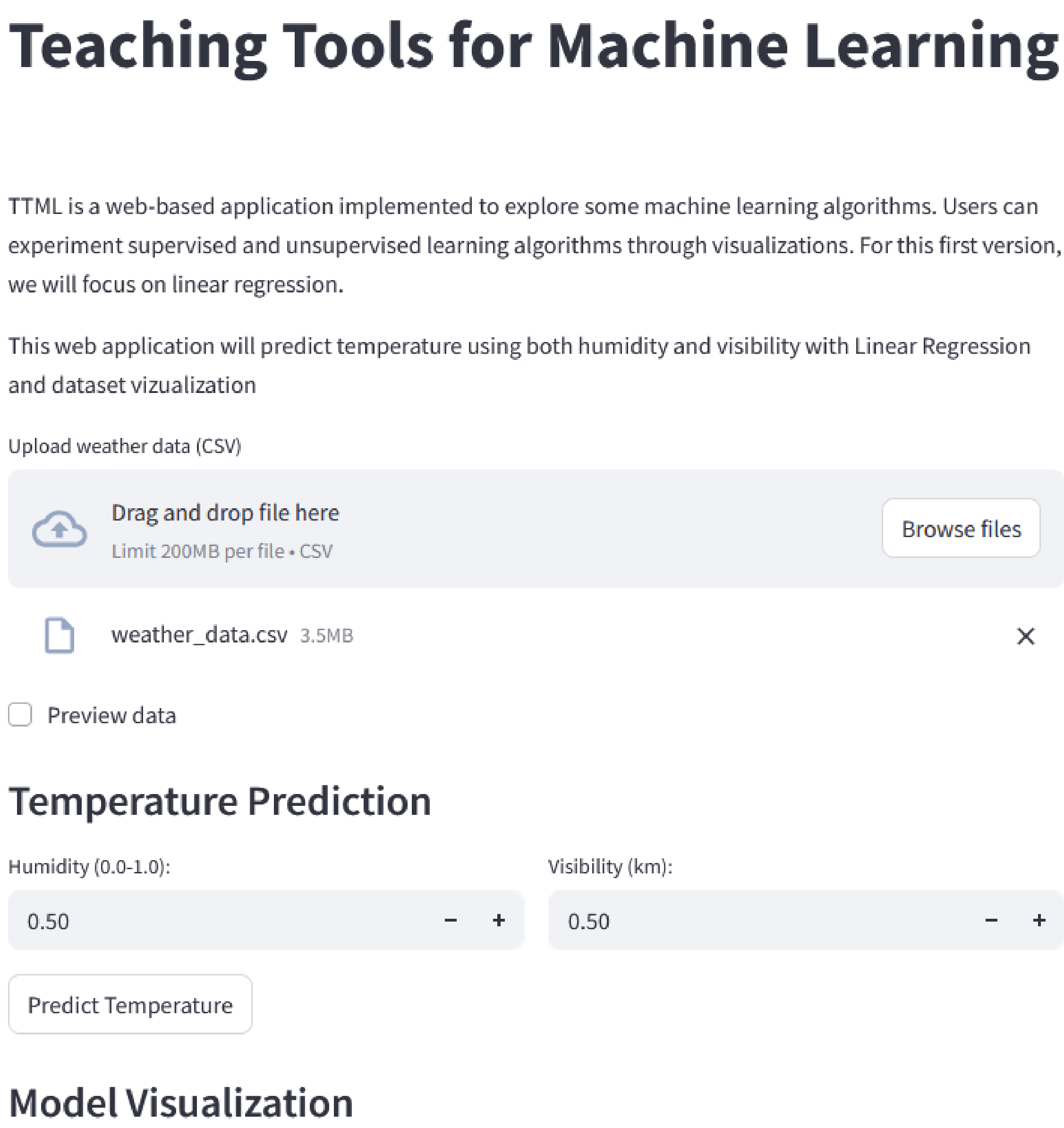
					Milestone #1				*Milestone #2*				*Milestone #3*				C-Day	
Phase	**Tasks**	**Complet**	**Curre**	**Assigned T**	**01/19**	**01/26**	**02/02**	**02/09**	**02/16**	**02/23**	**03/02**	**03/09**	**03/16**	**03/23**	**03/30**	**04/06**	**04/13**	**04/20**
	Requirement review	50%					5	10										
Project design	System Architecture	10%						10	4									
	Database design	0%						5	10	10								
	Requirements gathering	100%								10	10							
	Software Requirements Specification	100%		Kokou			6	6	10	5	5							
	Software Design Document	100%								10	10							
	Data Flow Diagram	100%									5	10	5					
Development	Implementation Git / Github	100%										8	5	10				
	Version control	100%		Kokou								8	10	20	20			
	Document updated design	100%													10	10		
	Testing	50%												8	5	20		
Final report	Final Report Package	100%														15		10
	Source Code Repository	100%		Kokou														10
	Video Demo	100%																
	Website	100%																5
	Total work hours			**377**	**20**	**22**	**26**	**31**	**24**	**35**	**30**	**26**	**20**	**38**	**35**	**45**	**0**	**25**

* formally define how you will develop this project including source code management

Legend	
Planned	
Delayed	
Number	Work: man hours

Figure 2 - Gantt Chart

9.2 screen mockups



Teaching Tools for Machine Learning

TTML is a web-based application implemented to explore some machine learning algorithms. Users can experiment supervised and unsupervised learning algorithms through visualizations. For this first version, we will focus on linear regression.

This web application will predict temperature using both humidity and visibility with Linear Regression and dataset vizualization

Upload weather data (CSV)

Drag and drop file here
Limit 200MB per file • CSV

Browse files

weather_data.csv 3.5MB

Preview data

Dataset

	Humidity	Visibility (km)	Temperature (C)
0	0.89	0.983	0.5235
1	0.86	0.983	0.5211
2	0.89	0.929	0.5531
3	0.83	0.983	0.5019
4	0.83	0.983	0.5174

Temperature Prediction

Teaching Tools for Machine Learning

TTML is a web-based application implemented to explore some machine learning algorithms. Users can experiment supervised and unsupervised learning algorithms through visualizations. For this first version, we will focus on linear regression.

This web application will predict temperature using both humidity and visibility with Linear Regression and dataset vizualization

Upload weather data (CSV)

Drag and drop file here
Limit 200MB per file • CSV

Browse files

No file uploaded. Using local dataset (weather_data.csv)...

☐ Preview data

Temperature Prediction

Humidity (0.0-1.0):

0.50 − +

Visibility (km):

0.50 − +

Predict Temperature

Model Visualization

Linear Regression Surface

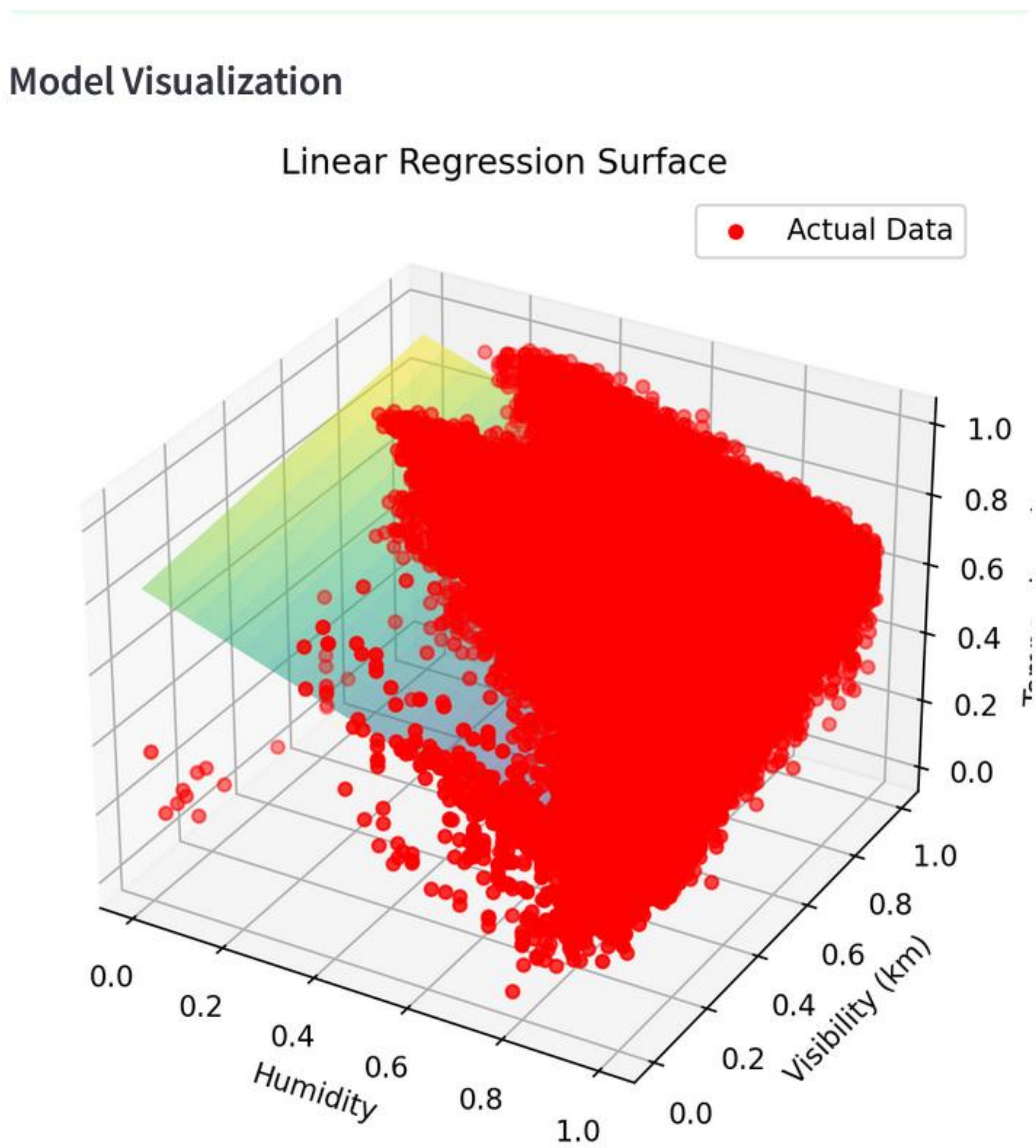


Figure 3 - screen mockups